

AD-A077 139

CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER
OPTIMAL LIST ORDER UNDER PARTIAL MEMORY CONSTRAINTS.(U)
MAR 79 Y C KAN , S M ROSS

F/G 12/1

N00014-77-C-0299

UNCLASSIFIED

ORC-79-4

NL

| OF |

AD
A077139



END

DATE

FILMED

12-79

DDC

ORC 79-4
MARCH 1979

LEVEL II

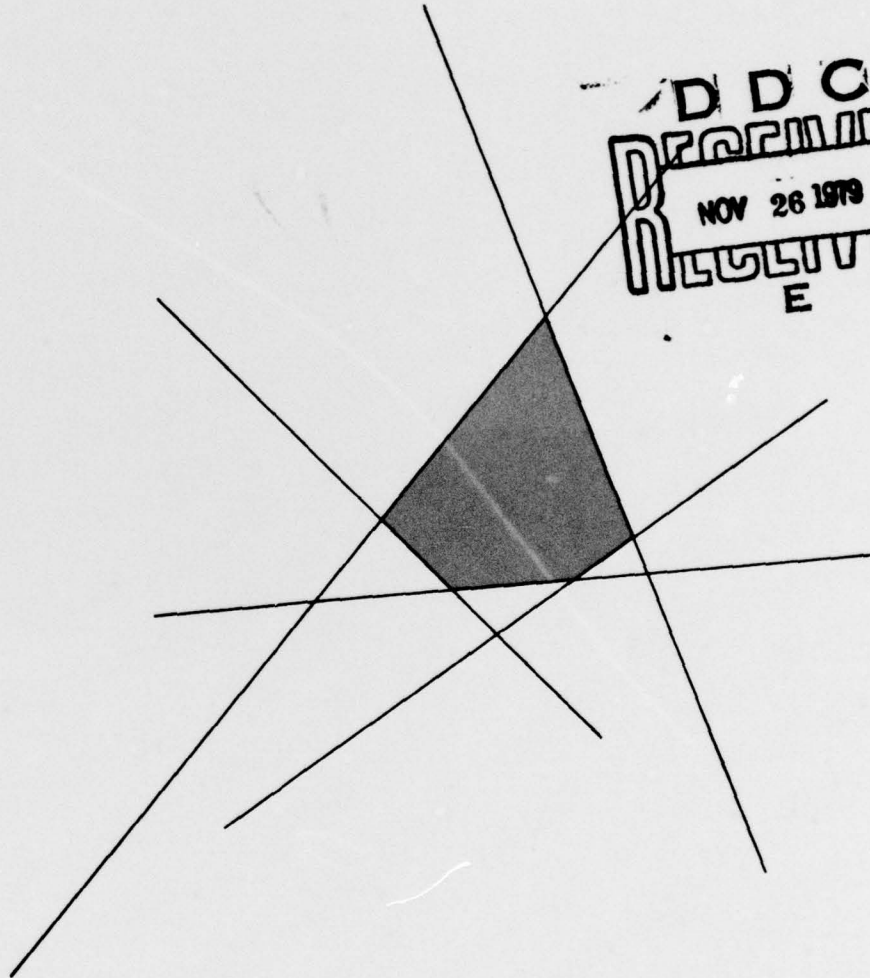
OPTIMAL LIST ORDER UNDER PARTIAL MEMORY CONSTRAINTS

by
YI C. KAN
and
SHELDON M. ROSS

P2

AD A 077139

DDC
RECEIVED
NOV 26 1979
E



This document has been approved
for public release and sale; its
distribution is unlimited.

UUC FILE COPY
OPERATIONS
RESEARCH
CENTER

UNIVERSITY OF CALIFORNIA • BERKELEY

79 11 26 039

12

OPTIMAL LIST ORDER UNDER PARTIAL MEMORY CONSTRAINTS

by

Yi C. Kan
Operations Research Center
University of California, Berkeley

and

Sheldon M. Ross
Department of Industrial Engineering
and Operations Research
University of California, Berkeley



This document has been approved
for public release and sale; its
distribution is unlimited.

MARCH 1979

ORC 79-4

This research has been partially supported by the Office of Naval Research under Contract N00014-77-C-0299 and the Air Force Office of Scientific Research (AFSC), USAF, under Grant AFOSR-77-3213B with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD FORM 1473
1 JAN 73

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

Suppose that we are given a set of n elements e_1, \dots, e_n which are to be arranged in some order. At each unit of time a request is made to retrieve one of these elements - e_i being requested (independently of the past) with probability P_i , $P_i \geq 0$, $\sum_{i=1}^n P_i = 1$. The cost of the retrieval is taken to be the ordered position of the element requested.

The problem of interest is to determine the optimal ordering so as to minimize the long run average cost. Clearly if the P_i were known the optimal ordering would simply be to order the elements in decreasing order of the P_i 's. In fact even if the P_i 's were unknown we could do as well asymptotically by ordering the elements at each unit of time in decreasing order of the number of previous requests for them. In this paper we first consider the case in which the only memory allowed at any time is the ordering of the elements at that time; in other words, the only type of reordering rules we allow are ones in which the reordered permutation of elements at any time is only allowed to depend on the present ordering and the position of the element requested. We show that the rule which always moves the requested element one closer to the front of the line minimizes the average position of the element requested among a wide class of rules for all probability vectors of the form $P_1 = p$, $P_2 = \dots = P_n = \frac{1-p}{n-1}$. In fact, we establish this under a stronger optimality condition - namely the criterion of stochastically minimizing the asymptotic position of the element requested.

We also consider the above problem under the proviso that additional memory is allowed. In particular we allow the decision-maker to utilize such rules as "only make a change (according to some preassigned rule) if the same element has been requested k times in a row." We show that as k approaches infinity we can do as well as if we knew the values of the P_i , and in addition we show that the convergence is monotone.

We then allow for the possibility of randomization. We first consider policies which at every unit of time follow some given rule with probability α and do nothing (make no reordering) with probability $1 - \alpha$; and show that their average costs are independent of α . However if we allow the randomization constant to be a function of the position of the element requested (one instance would be a policy which when the element selected is in position i moves it to the front with probability α_i and leaves the ordering unchanged with probability $1 - \alpha_i$) then the average cost depends on the sequence of randomization constants. Interestingly enough this is not the case for the one-closer rule whose average cost remains invariant under such randomization.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or special
A	

OPTIMAL LIST ORDER UNDER PARTIAL MEMORY CONSTRAINTS

by

Y. C. Kan and S. M. Ross

0. INTRODUCTION AND SUMMARY

Suppose that we are given a set of n elements e_1, \dots, e_n which are to be arranged in some order. At each unit of time a request is made to retrieve one of these elements - e_i being requested (independently of the past) with probability P_i , $P_i \geq 0$, $\sum_{i=1}^n P_i = 1$. The cost of the retrieval is taken to be the ordered position of the element requested.

The problem of interest is to determine the optimal ordering so as to minimize the long run average cost. Clearly if the P_i were known the optimal ordering would simply be to order the elements in decreasing order of the P_i 's. In fact even if the P_i 's were unknown we could do as well asymptotically by ordering the elements at each unit of time in decreasing order of the number of previous requests for them. However the problem becomes more interesting if we do not allow such memory storage as would be necessary for the above rule but rather restrict ourselves to a more limited memory storage. In [5] the case was considered where the only memory allowed at any time was the ordering of the elements at that time; in other words the only type of reordering rules allowed in [5] are ones in which the reordered permutation of elements at any time is only allowed to depend on the present ordering and the position of the element requested - we call such rules no-memory rules. A no-memory rule was said to be optimal in [5] if its average cost as a function of the probability vector \underline{P} is minimal among all rules for every probability vector \underline{P} having $0 < P_i < 1$, $i = 1, \dots, n$. Whereas it is not obvious

that an optimal rule exists it was conjectured in [5] that the rule which always moves the requested element one position closer to the front (called the transposition rule) is optimal. Though this conjecture was not proved it was shown in [5] that the transposition rule always has a smaller average cost than the one which moves the requested element to the front of the line.*

In Section 1 of the paper we consider the above problem under the premise that additional memory is allowed. In particular we allow the decision-maker to utilize such rules as "only make a change (according to some preassigned rule) if the same element has been requested k times in a row." We show that as k approaches infinity we can do as well as if we knew the values of the P_i , and in addition we show that the convergence is monotone.

In Section 2 we allow for the possibility of randomization. We first consider policies which at every unit of time follow some given (nonmemory and nonrandomized) rule with probability α and do nothing (make no re-ordering) with probability $1 - \alpha$; and show that their average costs are independent of α . However if we allow the randomization constant to be a function of the position of the element requested (one instance would be a policy which when the element selected is in position i moves it to the front with probability α_i and leaves the ordering unchanged with probability $1 - \alpha_i$) then the average cost depends on the sequence of randomization constants. Interestingly enough this is not the case for the (conjectured optimal) transposition rule whose average cost remains invariant under such randomization.

*If the present ordering is e_1, e_2, e_3, e_4 and element e_3 is requested then the transposition rule leads to the new ordering e_1, e_3, e_2, e_4 whereas the front of the line rule leads to e_3, e_1, e_2, e_4 .

In the final section we consider the original model where the only rules allowed are ones whose reordering is based on the present ordering and the position of the element requested. We show that the transposition rule is optimal among a wide class of rules for all probability vectors of the form $P_1 = p$, $P_2 = \dots = P_n = \frac{1-p}{n-1}$. In fact we establish this under a stronger optimality condition - namely the criterion of stochastically minimizing the asymptotic position of the element requested.

1. K-IN-A-ROW POLICIES

Consider any rule R which after each request reorders the list solely as a function of the present ordering and the position of the element requested and suppose now that we are allowed to follow the policy where we only make a change in the list order (according to rule R) if the same element has been requested k times in a row. (Such policies would require two additional counters of memory space - one for keeping track of the last element requested and the other keeping track of the number of times in a row it had been requested.) Once an element has been requested k times in a row we reorder the list according to R and then start over again as far as waiting for another run of k identical requests.

The sequence of list orderings which result under the above policy can most easily be analyzed as a semi-Markov process with the state at any time being the ordering at that time and the epochs of transition being the times at which a run of k identical requests have occurred. We start by computing the probability that any given run of k identical requests were all requests for element i .

Proposition 1.1:

Given a sequence of independent multinomial trials - each resulting in outcome i with probability p_i , $\sum_{i=1}^n p_i = 1$. Then the probability that a run of k_1 successive trials all resulting in outcome number 1 occurs before any run of k_i successive i outcomes, $i = 2, \dots, n$ equals

$$\frac{p_1^{k_1}(1 - p_1)/(1 - p_1^{k_1})}{\sum_{i=1}^n p_i^{k_i}(1 - p_i)/(1 - p_i^{k_i})}.$$

Proof:

We first compute the expected number of coin tosses, call it $E[T]$, until a run of N successive heads occur when the tosses are independent and each lands on heads with probability p . By conditioning on the time of the first nonhead we obtain

$$E[T] = \sum_{j=1}^N (1-p)p^{j-1}(j + E[T]) + Np^N.$$

Solving the above for $E[T]$ yields

$$E[T] = N + \frac{(1-p)}{p^N} \sum_{j=1}^N jp^{j-1}$$

and, simplifying, we obtain

$$E[T] = \frac{1 + p + \dots + p^{N-1}}{p^N} = \frac{(1 - p^N)}{p^N(1 - p)}.$$

Now consider the (infinite) sequence of multinomial trials as specified in the statement of the proposition. Let us say that an i -success occurs whenever we obtain a run of k_i successive i outcomes. Then by renewal theory the rate of i -successes is just 1 divided by the expected time between i -successes and so the proportion of successes that are i successes is (with probability 1)

$$\frac{1}{E(T_i)} / \sum_{j=1}^n \frac{1}{E(T_j)} = \frac{p_i^{k_i}(1 - p_i) / (1 - p_i^{k_i})}{\sum_{j=1}^n \frac{p_j^{k_j}(1 - p_j) / (1 - p_j^{k_j})},$$

where T_i is the time between i -successes.

But each time a success (that is an i -success for any i) occurs everything restarts itself and so the limiting proportion of successes that are of type i must also equal the probability that an i -success occurs before any j -success, $j \neq i$. ||

Now in a semi-Markov process if we let π_i denote the limiting probability of being in state i for the embedded Markov chain which looks at the process only when transitions occur and we let μ_i denote the mean time until the next transition when in state i then the limiting proportion of time the state is i equals $\pi_i \mu_i / \sum \pi_j \mu_j$. Hence since in our problem the mean time spent in any state is constant--it is just the expected time to obtain a run of k requests for the same element--it follows that the limiting proportion of time spent in each state is equal to the limiting probabilities for the embedded Markov chain which only considers the successive orderings when transitions (i.e., runs of k in a row) occur.

Thus it follows that the performance of the policy which uses rule R only when there have been k requests in a row for the same element is exactly the same as the performance of rule R in the case where the request probabilities are no longer P_1, \dots, P_n but rather are now given by $P_1^{(k)}, \dots, P_n^{(k)}$ where

$$P_i^{(k)} = \frac{P_i^k (1 - P_i) / (1 - P_i^k)}{\sum_{j=1}^n P_j^k (1 - P_j) / (1 - P_j^k)}.$$

The next lemma shows that as k increases the proportion of requests (in the embedded chain) for the element having the largest request probability increases to 1; among the remaining requests the proportion of those that are

for the element having the second largest request probability also increases to 1, etc.

Lemma 1.2:

If $P_1 > P_2 > \dots > P_n$ then

$$\frac{P_i^{(k)}}{\sum_{j=i}^n P_j^{(k)}} \rightarrow 1 \text{ as } k \rightarrow \infty, i = 1, \dots, n.$$

Proof:

We must show that

$$\frac{P_i^k(1 - P_i)/(1 - P_i^k)}{\sum_{j=i}^n P_j^k(1 - P_j)/(1 - P_j^k)} \rightarrow 1 \text{ as } k \rightarrow \infty.$$

By dividing numerator and denominator by $P_i^k/(1 - P_i^k)$ we see that it suffices to show that

$$\left(\frac{P_j}{P_i}\right)^k \left(\frac{1 - P_i^k}{1 - P_j^k}\right) \rightarrow 0 \text{ as } k \rightarrow \infty \text{ when } P_i > P_j.$$

As the above equals $\frac{(1/P_i)^k - 1}{(1/P_j)^k - 1}$ the result follows Lemma 1.3. ||

Lemma 1.3:

$$\frac{a^x - 1}{b^x - 1} \rightarrow 0 \text{ as } x \rightarrow \infty \text{ when } b > a > 1.$$

Proof:

The derivative of $\frac{a^x - 1}{b^x - 1}$ will be negative if

$$\frac{a^x \ln a}{a^x - 1} < \frac{b^x \ln b}{b^x - 1}$$

but this follows from the fact that $\frac{b^x \ln b}{b^x - 1}$ is an increasing function of b when $b > 1$, which is easily established upon differentiation. ||

Theorem 1.4:

Let R be any rule which moves the element requested strictly closer to the front (unless it is already in the initial position in which case it remains there) and leaves the relative ordering of the other elements unchanged. Then under the policy which follows rule R only when there have been k requests in a row for the same element the proportion of time the element with the j^{th} largest request probability is in position j goes to 1 as k goes to ∞ .

Proof:

Suppose the elements are numbered so that $P_1 > P_2 > \dots > P_n$. By Lemma 1.2 it follows that the proportion of reorderings that result in element 1 being moved closer to the front of the line goes to 1 as k becomes large. Hence it follows that the proportion of time that element 1 is in position 1 also goes to 1 as k gets larger. The remainder of the argument is similar. ||

Thus we see that as k goes to infinity the proportion of time that the ordered list corresponds to the optimal ordering when the P_i 's are known goes to 1. Hence the average cost under any of the policies specified in Theorem 1.4 converges to what the average cost would be if the P_i 's are known - namely $\sum_{i=1}^n iP_{(i)}$ where $P_{(i)}$ is the i^{th} largest of P_1, \dots, P_n . From the results of Lemma 1.2 it would also seem reasonable that this convergence would be monotone. We will verify this monotone convergence for the easiest rule to analyze, namely the one which moves the requested element to the front of the line.

Now under the front of the line rule if the elements have probabilities P_1, \dots, P_n then the expected position (with respect to the limiting distribution) of the element requested can be expressed as

$$\begin{aligned} \text{Average Cost} &= \sum_{j=1}^n P_j E\{\text{position of element } j\} \\ &= \sum_{j=1}^n P_j \sum_{i \neq j} P\{i \text{ precedes } j\} \\ &= \sum_{j=1}^n \sum_{i \neq j} P_j P_i / (P_i + P_j) \end{aligned}$$

where we have used the fact that $P\{i \text{ precedes } j\}$ is the probability that after a long time the most recent request for either i or j was for i , which is easily seen to equal $P_i / (P_i + P_j)$. The above formula was derived in [1], [3], [4] and [5].

Hence we want to show that

$$\sum_{j=1}^n \sum_{i \neq j} P_j^{(k)} P_i^{(k)} / (P_i^{(k)} + P_j^{(k)}) \rightarrow \sum_{j=1}^n P_j \sum_{i \neq j} P_i / (P_i + P_j) \quad \text{as } k \rightarrow \infty.$$

To prove this we first introduce the concepts of majorization and Schur functions. We say the vector $\underline{x} = (x_1, \dots, x_n)$ majorizes the vector $\underline{y} = (y_1, \dots, y_n)$, written as $\underline{x} \underset{m}{\geq} \underline{y}$ if

$$\sum_{i=1}^j x_{(i)} \geq \sum_{i=1}^j y_{(i)}, \quad j = 1, \dots, n-1$$

and

$$\sum_{i=1}^n x_{(i)} = \sum_{i=1}^n y_{(i)}$$

where $x_{(i)}$, $y_{(i)}$ are the i^{th} largest values of x_1, \dots, x_n and y_1, \dots, y_n respectively.

The symmetric function f is said to be a Schur concave function if $f(\underline{x}) \leq f(\underline{y})$ whenever $\underline{x} \underset{m}{\geq} \underline{y}$. The following criterion for determining if a function is Schur concave is due to Ostrowski.

Theorem: (Ostrowski)

A differentiable symmetric function f is Schur concave if and only if

$$(x_1 - x_2) \left(\frac{\partial f(\underline{x})}{\partial x_1} - \frac{\partial f(\underline{x})}{\partial x_2} \right) \leq 0 \quad \text{for all } \underline{x}.$$

Proof:

See [2], p. 47.

We are now ready for

Proposition 1.5:

The function

$$H(\underline{x}) = \sum_{j=1}^n \sum_{i \neq j} x_i x_j / (x_i + x_j), \quad x_i \geq 0$$

is a Schur concave function.

Proof:

$$\begin{aligned} \frac{\partial H(\underline{x})}{\partial x_1} &= \sum_{i \neq 1} x_i^2 / (x_i + x_1)^2 + \sum_{j \neq 1} x_j^2 / (x_1 + x_j)^2 \\ &= 2 \sum_{i \neq 1} x_i^2 / (x_i + x_1)^2. \end{aligned}$$

Therefore

$$\begin{aligned} (x_1 - x_2) \left(\frac{\partial H(\underline{x})}{\partial x_1} - \frac{\partial H(\underline{x})}{\partial x_2} \right) &= 2(x_1 - x_2) \left[\sum_{i \neq 1} x_i^2 / (x_1 + x_i)^2 - \sum_{i \neq 2} x_i^2 / (x_2 + x_i)^2 \right] \\ &= 2(x_1 - x_2) \left[\sum_{i=3}^n \left[\left(\frac{x_i}{x_1 + x_i} \right)^2 - \left(\frac{x_i}{x_2 + x_i} \right)^2 \right] + \right. \\ &\quad \left. \left(\frac{x_2}{x_1 + x_2} \right)^2 - \left(\frac{x_1}{x_1 + x_2} \right)^2 \right]. \end{aligned}$$

As we see that this is nonpositive, the result follows from the Ostrowski theorem. ||

Theorem 1.6:

If the front of the line rule is only utilized when the same element has been requested k times in a row then the average cost of this policy, namely $\sum_{j=1}^n \sum_{i \neq j} p_j^{(k)} p_i^{(k)} / (p_j^{(k)} + p_i^{(k)})$, is a decreasing function of k .

Proof:

This result will follow from the previous proposition upon showing that

$$p_1^{(k+1)}, \dots, p_n^{(k+1)} \geq_m p_1^{(k)}, \dots, p_n^{(k)}.$$

Assume without loss of generality that the elements are numbered so that

$p_1 \geq p_2 \geq \dots \geq p_n$ which will imply the same orderings for the vector $\underline{p}^{(k)}$.

Let

$$\lambda_i^{(\ell)} = \frac{p_i^{(\ell)}}{\sum_{j=i}^n p_j^{(\ell)}}, \quad i = 1, \dots, n$$

and note that from Lemma 1.2, we have that

$$\lambda_i^{(k+1)} \geq \lambda_i^{(k)}, \quad i = 1, \dots, n.$$

As it is easy to establish (by induction on j) that

$$\prod_{i=1}^j (1 - \lambda_i^{(\ell)}) = \sum_{i=j+1}^n p_i^{(\ell)}, \quad j = 1, \dots, n$$

the result follows.

2. RANDOMIZATION

The reason that the k-in-a-row policies do better than the no-memory reordering rules is that these latter rules make changes too frequently. For instance if we were allowed perfect memory then the best rule would be to order the elements in decreasing order of the number of requests for them. Hence, after a while, reorderings become infrequent - for instance if $n = 4$ and the total number of requests for elements 1 through 4 are at present 20, 60, 10, 80 then the optimal ordering would be e_4, e_2, e_1, e_3 and would remain so for at least the next 10 periods regardless of the elements requested in this time span.

Another approach to slowing down changes in list order is to allow for randomized policies. In particular consider any no-memory rule R and consider the policy which when the element requested is in position i follows the dictates of rule R with probability α_i and leaves the present ordering unchanged with probability $1 - \alpha_i$, for given α_i , $0 \leq \alpha_i \leq 1$, $i = 1, \dots, n$. We first note that if the randomization value α_i is the same for all i , say $\alpha_i \equiv \alpha$, then the average cost for the randomized policy is the same as that of the original rule R .

Proposition 2.1:

If $\alpha_i \equiv \alpha$, $i = 1, \dots, n$ then the average cost of the randomized policy is independent of α .

Proof:

We can analyze the sequence of orderings as a semi-Markov process where a transition occurs whenever the outcome of the randomization results in rule R being followed. As this occurs with probability α independent

of the particular order it follows that the mean time spent in each state during a visit is $1/\alpha$ for every state. Hence the limiting probabilities are exactly the same as those of the embedded Markov chain which considers the orderings only at times where R is followed. As these limiting probabilities are clearly the same as when $\alpha = 1$ the result follows. ||

In general the average cost of a randomized policy based on the rule R will depend on the values of the α_i . However an interesting exception is when R is taken to be the (conjectured optimal) transposition rule. For this case we first note the following lemma which was also proved in [5] for the special case $\alpha_i \equiv 1$.

Lemma 2.2:

For the randomized policy based on the transposition rule and using randomization constants α_i , $i = 1, \dots, n$,

$$(2.1) \quad P_{i_{j+1}} \Pr(i_1, \dots, i_j, i_{j+1}, \dots, i_n) = P_{i_j} \Pr(i_1, \dots, i_{j+1}, i_j, \dots, i_n)$$

where i_1, \dots, i_n is any permutation of $1, 2, \dots, n$ and $\Pr(i_1, \dots, i_n)$ is the stationary probability that the list order is (i_1, \dots, i_n) given that the stated policy is employed.

Proof:

By multiplying both sides of Equation (2.1) by α_{j+1} we see that (2.1) is equivalent to stating that rate at which the Markov chain goes from any state s to s' is equal to the rate at which it goes from s' to s ; or in other words, it states that the Markov chain is time reversible. Now it is well-known that a necessary and sufficient condition

for time reversibility is that for any sequence of states $s, s^1, s^2, \dots, s^i, s$ the transition probabilities must satisfy $P_{s, s^1} P_{s^1, s^2} \dots P_{s^i, s} =$

$P_{s, s^i} \dots P_{s^1, s}$ that this is the case is easily verified for this particu-

lar model. (For instance if $n = 3$ and the sequence of states is

$(1, 2, 3), (2, 1, 3), (2, 3, 1), (3, 2, 1), (3, 1, 2), (1, 3, 2), (1, 2, 3)$ the product of

the transition probabilities going from left to right is

$\alpha_2 p_2 \alpha_3 p_3 \alpha_2 p_3 \alpha_3 p_1 \alpha_2 p_1 \alpha_3 p_2 = \alpha_2^3 \alpha_3^3 p_1^2 p_2^2 p_3^2$ whereas in the reverse direction

it is $\alpha_3 p_3 \alpha_2 p_3 \alpha_3 p_2 \alpha_2 p_2 \alpha_3 p_1 \alpha_2 p_1 = \alpha_2^3 \alpha_3^3 p_1^2 p_2^2 p_3^2$. ||

Since the stationary probabilities are obtained from the set of equations (2.1) which do not depend on the α_i , we have

Theorem 2.3:

The average cost of any randomized policy based on the transposition rule is independent of the randomization constants α_i .

3. TRANSPOSITION RULE OPTIMALITY WHEN $P_1 = p$, $P_i = \frac{1-p}{n-1}$, $i = 2, \dots, n$

In this section we shall suppose that $P_1 = p$, $P_2 = \dots = P_n = \frac{1-p}{n-1} = q$. In general under any rule, the average cost can be obtained by analyzing the Markov chain of $n!$ states where the state at any time corresponds to the ordering at that time. However for the P_i of the form above, as all of the elements 2 through n are identical (as they have the same probability of being requested) we can obtain the average cost by analyzing the much simpler Markov chain of n states with the state being the position of element 1.

Consider the following restricted class of rules which when an element is requested and found in position i , move the element to position j_i and leave the relative positions of the other elements unchanged. In addition we suppose that $j_i < i$ for $i > 1$, $j_1 = 1$ and $j_i \geq j_{i-1}$, $i = 2, \dots, n$. The set $\{j_i, i = 1, \dots, n\}$ characterizes a rule in this class.

For a given rule in the above class let

$$k(i) = \max \{l : j_{i+l} \leq i\}.$$

In other words, for any i , an element in any of the positions $i, i+1, \dots, i+k(i)$ will, if requested, be moved to a position less than or equal to i .

For a specified rule in the above class let us denote the stationary probabilities when this rule is employed by

$$\pi_i = \Pr\{e_1 \text{ is in position } i\}, i = 1, \dots, n$$

$$S_i = \sum_{j=i+1}^n \pi_j = \Pr\{e_1 \text{ is in a position } > i\}, i = 0, 1, \dots, n-1.$$

Before writing down the steady state equations it may be worth noting the following:

- (i) Any element moves toward the back of the list at most one position at a time.
- (ii) If an element is in position i and neither it nor any of the elements in the following $k(i)$ positions are requested it will remain in position i .
- (iii) Any element in one of the positions $i, i+1, \dots, i+k(i)$ will be moved to a position $\leq i$ if requested.

The steady state probabilities can now easily be seen to be:

$$S_i = S_{i+k(i)} + (S_i - S_{i+k(i)})(1-p) + (S_{i-1} - S_i)qk(i)$$

or

$$S_i = a_i S_{i-1} + (1-a_i) S_{i+k(i)}, i = 1, \dots, n-1$$

$$(3.1) \quad S_0 = 1, S_n = 0$$

where

$$(3.2) \quad a_i = \frac{qk(i)}{qk(i) + p}.$$

Now consider a special rule of the above class, namely the transposition rule which has $j_i = i - 1$, $i = 2, \dots, n$, $j_1 = 1$. Let the corresponding S_i be denoted by S'_i for the transposition rule. Then from Equation (3.1) we have, since $k(i) = 1$, that

$$S'_i = \frac{qS'_{i-1} + pS'_{i+1}}{p + q}$$

or, equivalently,

$$S'_{i+1} - S'_i = \frac{q}{p} (S'_i - S'_{i-1})$$

which, iterating, implies

$$S'_{i+r} - S'_{i+r-1} = \left(\frac{q}{p}\right)^r (S'_i - S'_{i-1}).$$

Summing the above equations from $r = 1, \dots, r$ we obtain

$$S'_{i+r} - S'_i = (S'_i - S'_{i-1}) \left[\frac{q}{p} + \dots + \left(\frac{q}{p}\right)^r \right], \quad i + r \leq n.$$

Now consider any other rule R of the considered class and let $k(i)$ be as defined for that rule. Now from the above we see that for the transposition rule

$$S'_{i+k(i)} - S'_i = (S'_i - S'_{i-1}) \left(\frac{q}{p} + \dots + \left(\frac{q}{p}\right)^{k(i)} \right)$$

or, equivalently

$$(3.3) \quad S'_i = b_i S'_{i-1} + (1 - b_i) S'_{i+k(i)}$$

where

$$(3.4) \quad b_i = \frac{(q/p) + \dots + (q/p)^{k(i)}}{1 + (q/p) + \dots + (q/p)^{k(i)}}, \quad i = 1, \dots, n-1$$

and $S'_n = 0$, $S'_0 = 1$.

We are now ready to prove

Theorem 3.1:

If $p \geq 1/n$, then $S'_i \leq S_i$ for all i .

If $p \leq 1/n$, then $S'_i \geq S_i$ for all i .

Proof:

Consider the case $p \geq 1/n$ which is equivalent to $p \geq q$ and note that in this case

$$a_i = 1 - \frac{1}{1 + \frac{k(i)}{p} q} \geq 1 - \frac{1}{1 + q/p + \dots + (q/p)^{k(i)}} = b_i.$$

Now define a Markov chain with states $0, 1, \dots, n$ and transition probabilities

$$(3.5) \quad P_{ij} = \begin{cases} c_i & \text{if } j = i-1 \\ 1 - c_i & \text{if } j = i+k(i) \end{cases} \quad i = 1, \dots, n-1.$$

Let f_i denote the probability that this Markov chain ever enters state 0 given that it starts in state i . Then f_i satisfies

$$f_i = c_i f_{i-1} + (1 - c_i) f_{i+k(i)} \quad i = 1, \dots, n-1$$

$$f_0 = 1, \quad f_n = 0.$$

Hence, as it is well known that the above set of equations has a unique solution, it follows from (3.1) that if we take c_i equal to a_i for all i , then f_i will equal the S_i of rule R , and if we let $c_i = b_i$, then f_i equals S'_i . Let $X_r(\underline{a})$ and $X_r(\underline{b})$ denote the state at time r of the Markov chain defined by (3.5) when c_i equals respectively a_i and b_i . Now, as $P\{X_1(\underline{a}) \geq j \mid X_0(\underline{a}) = i\}$ and $P\{X_1(\underline{b}) \geq j \mid X_0(\underline{b}) = i\}$ are both increasing in i , for all j , and as

$$P\{X_1(\underline{a}) \geq j \mid X_0(\underline{a}) = i\} \leq P\{X_1(\underline{b}) \geq j \mid X_0(\underline{a}) = i\}$$

for all j , it can be shown (see Theorem 4 of [6]) that

$$P\{X_r(\underline{a}) \geq j \mid X_0(\underline{a}) = i\} \leq P\{X_r(\underline{b}) \geq j \mid X_0(\underline{b}) = i\}.$$

Hence,

$$P\{X_r(\underline{a}) = 0\} \geq P\{X_r(\underline{b}) = 0\} \text{ for all } r$$

implying that

$$S_i \geq S'_i.$$

When $p \leq 1/n$, then $a_i \leq b_i$, and the above inequality is reversed. ||

Thus for all rules in the considered class, the asymptotic position of element 1 will be stochastically larger (smaller) than it would be under the transposition rule when $p \geq 1/n$ ($p \leq 1/n$).

Now consider any cost function whose cost of requesting an element in position i --call it $g(i)$ --is an increasing function of i . Letting X denote the (asymptotic) position of element 1, we have that the average expected cost can be expressed as

$$E[\text{cost}] = pE[g(X)] + (1 - p) \frac{E[g(1) + \dots + g(n) - g(X)]}{n - 1}.$$

The above follows by conditioning on whether or not element 1 is requested and then noting that if element 1 is not requested, then any of the remaining $n - 1$ elements are equally likely to be. Hence,

$$E[\text{cost}] = \left(p - \frac{1 - p}{n - 1} \right) E[g(X)] + (1 - p) \frac{(g(1) + \dots + g(n))}{n - 1};$$

and thus if $p \geq 1/n$, the expected cost is minimized by minimizing $E[g(X)]$ and if $p \leq 1/n$ by maximizing $E[g(X)]$. But as stochastically minimizing (maximizing) X is equivalent to minimizing (maximizing) $E[g(X)]$, for every increasing function g , it follows that the expected cost is minimized by the transposition rule.

REFERENCES

- [1] Burville, P. J. and J. F. C. Kingman, "On a Model for Storage and Search," Journal of Applied Probability, Vol. 10, pp. 697-701, (1973).
- [2] Hardy, G. H., J. E. Littlewood and G. Polya, INEQUALITIES, Cambridge University Press, Cambridge, MA, 1952.
- [3] Hendricks, W. J., "The Stationary Distribution of an Interesting Markov Chain," Journal of Applied Probability, Vol. 9, pp. 231-233, (1972).
- [4] McCabe, J., "On Serial Files with Relocatable Records," Operations Research, Vol. 12, pp. 601-618, (1965).
- [5] Rivest, R., "On Self-Organizing Sequential Search Heuristics," Communications of the ACM, Vol. 19, No. 2, (February 1976).
- [6] Veinott, A. F., "Optimal Policy in a Dynamic, Single Product, Nonstationary Inventory Model with Several Demand Classes," Operations Research, Vol. 13, pp. 761-778, (1965).

